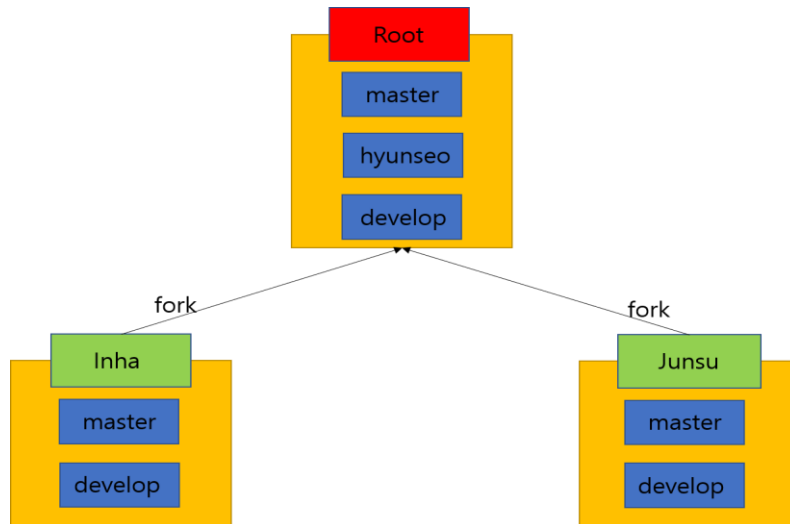
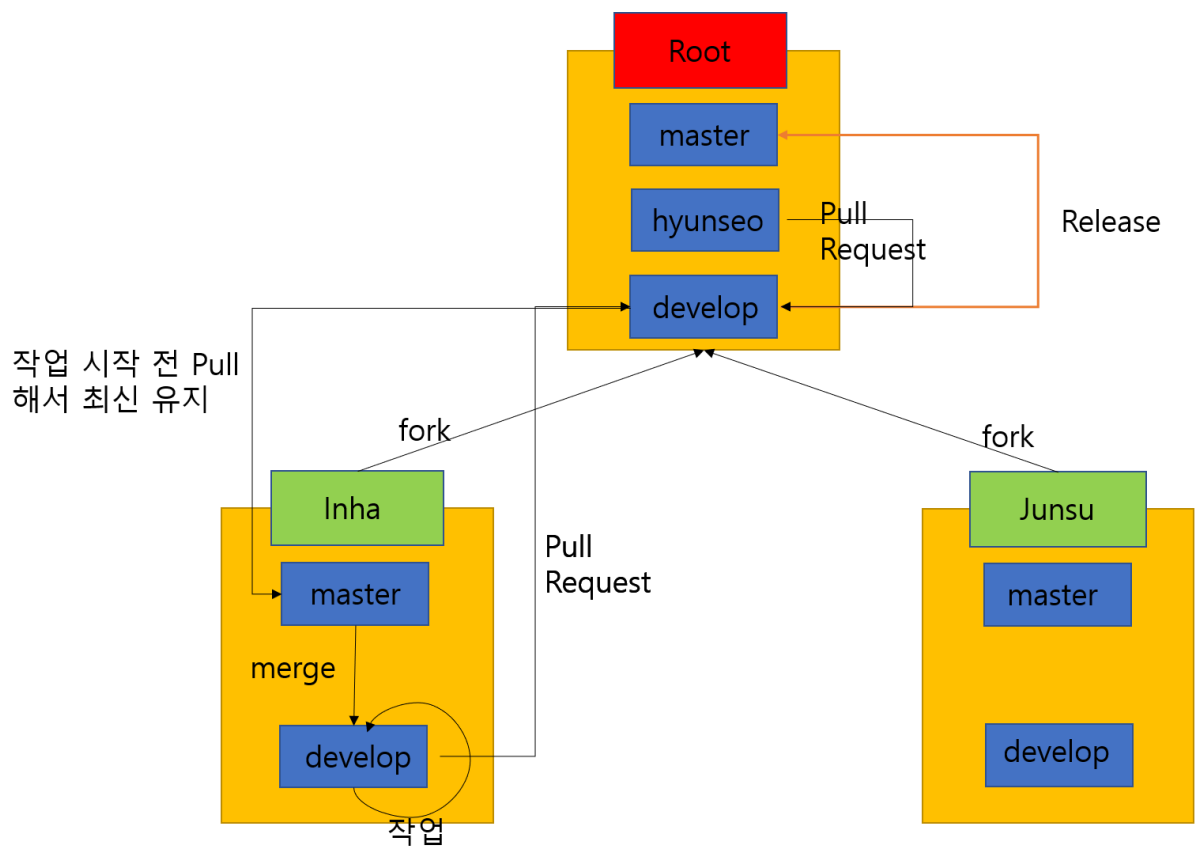


Git 사용

구조



정책



Fork된 개인 Repository의 develop branch에서 개발해야 한다.

Fork된 개인 Repository를 PC로 가져오기

```
git clone 개인 저장소 주소
```

develop branch 생성

```
git checkout -b develop
```

원격 저장소 설정

- 개인 원격 저장소

```
git remote add origin 개인 저장소 주소
```

- Pull Request를 날릴 Root 주소

```
git remote add upstream https://github.com/Parkhyunseo/CellLove
```

설정이 다 되었으면, Fork된 개인 Repository의 develop branch에서 개발

개발 완료를 했거나 특정 파일을 git이 tracking하게 하려면

```
git add 파일명 또는 .
```

Task를 완료하여 커밋을 하려고 한다.

```
git commit -m "내용"
```

커밋 메시지 작성은 <https://item4.github.io/2016-11-01/How-to-Write-a-Git-Commit-Message/> 를 참고한다.

Fork된 원격저장소에 파일을 올려야 한다.

```
git push origin develop
```

push가 완료 되었다면, github로 들어가 Pull request를 날린다. 여기서 PR(Pull Request)는 개인 Repository develop branch에서 Upstream(Root) develop branch로 해야한다.

다음 Task를 진행 할 때는 master로 checkout 후 pull upstream을 하여 최신을 유지하고 develop으로 다시 돌아가 merge 후 Task를 시작한다.

```
git checkout master
```

```
git pull upstream develop
```

```
git checkout develop
```

```
git merge master
```

칸반 보드 정책

TODO

백로그(개발 해야할 추상적인 내용들)가 들어간다. 백로그는 회의 시간에 작성된다. 백로그를 바탕으로 구체적인 내용이 나오고 개발에 들어갈 준비가 되면 Issue를 생성하게 되고 Issue도 TODO에 위치한다. Issue를 작성하면 자동으로 추가된다.

In Progress

회의를 통해 TODO라인에서 오늘 할당 받아 진행할 Task를 가져다 놓는 곳이다.

Needs Reviews

In Progress라인에서 개발을 완료하여 Pull Request를 날린 Task를 이곳에 위치한다. 여기에 위치한 Task들은 코드리뷰를 해줘야 한다. 적을 코멘트가 없으면 확인이라는 코멘트나 이모지를 남긴다.

Reviews Approved

Needs Reviews라인에서 Review가 끝난다면, 이 라인에 오게 된다. 관리자는 이곳의 Task를 확인 후 root의 develop branch에 merge를 수행한다.

Done

위의 모든 라인을 지났다면 Done으로 오게 된다. Done은 Issue tracking을 위해서 필요하다.